

# Speeding up SPS Slow Extraction Simulations with PTC Maps at Arbitrary Order

ICFA Mini-Workshop on Slow Extraction 2019

F.M. Velotti, H. Bartosik, Y. Dutheil, M. Fraser,  
B. Goddard, V. Kain

July 23<sup>rd</sup>, 2019

## Introduction

From element-by-element to maps tracking

Tracking and maps validation - SPS slow extraction

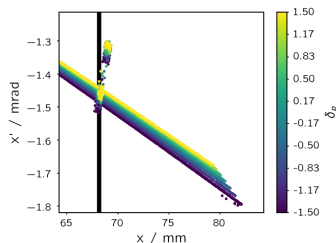
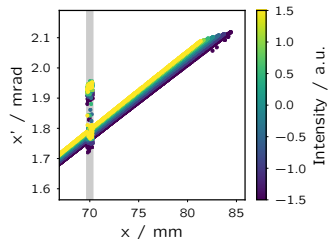
- Normal SPS slow extraction

- Separatrix folding slow extraction

Application examples

Conclusions and outlook

- Slow extraction simulations are quite complicated by design
- We like to sit on top of resonances, use strong non-linear elements, compute losses, evaluate multi-turn effects from scattering, using exotic devices...etc.
- Of course, then we also want to run scans of parameters and maybe even automatically optimise them
- All this means that we need a very efficient way to simulate the slow extraction process without loosing in accuracy!
- For the SPS slow extraction simulations we have been using MADX thin tracking so far
- MADX is very good to do accurate simulations and to monitor the full accelerator aperture model  $\Rightarrow$  **thin lenses tracking element by element**
- It has been interfaced with pycollimate (for crystal and scattering) but, overall, it is quite slow (MADX+pycollimate was not intended for multi-turn simulations)!
- We have been looking around to try to find a tracking code/methodology to speed things up as it is impossible to run scans otherwise!



- A first idea was trying to branch off from element-by-element tracking as very time consuming...but we could not go to direct one-turn map approach as we need to “unconventionally” act on the beam every turn and at different locations
- A compromise could be reducing the full machine to a few “sectors”  $\Rightarrow$  need a way to compute the transformation through a full sector...many elements and **strong non-linearities!** ...not a simple task...
- A first idea was to use truncated power (Taylor) series  $\Rightarrow$  this compromises symplecticity if elements with  $L > 0$  are used
- One could argue that we maybe don't need to be symplectic as we track only for a few turns...**but this is not the topic of this talk**
- Anyway, using available symplectic integrators symplecticity can be nevertheless ensured
- In short summary: using a symplectic integrator code and using power series expansion around the closed orbit (or reference trajectory), symplectic maps (well, numerically symplectic) up to arbitrary orders can be obtained to finally use them for tracking
- **This is what we have implemented for the SPS slow extraction simulations**

- The Polymorphic Tracking Code (PTC) [1] of E. Forest [2] gives access to different models for symplectic integration - possibility to use up to 6<sup>th</sup> order integrators (Yoshida method) [3] and finally also incorporates Truncated Power Series Algebra (TPSA) packages
- In fact, using a MADX lattice, it is then possible to use PTC (we are using from the MADX interface) for normal form analysis, for example, or, as we did, to **extract sector or one-turn maps!**
- What comes out are transfer maps obtained from tracking of the identity around the CO
- The number of maps is directly linked to the accuracy of the tracking...and this depends on the machine configuration under analysis



# Truncated power series generation

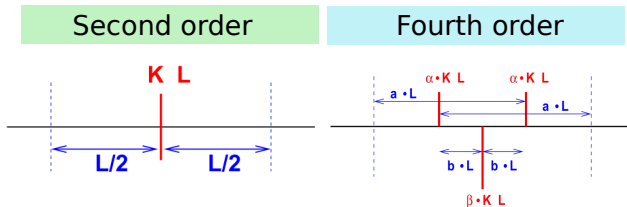
- For example, in PTC one can use the classic “drift-kick-drift” integrator when the element Hamiltonian is separable, i.e.  $H = H_1 + H_2$
- This is nothing else than splitting in thin kicks a previous tick element
- The simplest case is given by a second order integrator

$$S_2(dz) = \exp\left(-\frac{dz}{2} H_1\right) \exp(-dz H_2) \exp\left(-\frac{dz}{2} H_1\right) \quad (1)$$

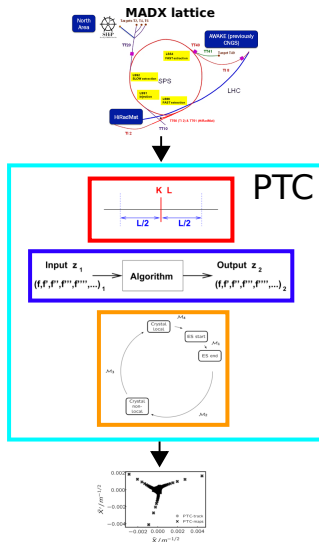
- Then this can be done iteratively, obtaining higher order integrators in this way...for example from a second order (1 kick) one can construct a 4<sup>th</sup> order integrator (3 times 1 kick = 3 kicks) [3]:

$$S_4(dz) = S_2(x_1 dz) S_2(x_0 dz) S_2(x_1 dz) \quad (2)$$

- Iterative formulas are available [4] to construct high-order integrators



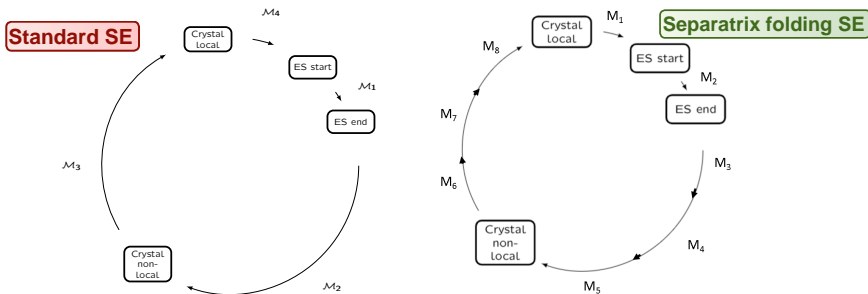
- Once the lattice is converted with a chosen number of integration steps, integrator model and order, we can extract the needed maps
- Tracking the identity through the lattice, the expansion in Taylor series of the map for each defined sector around the reference CO can be obtained**
- This can be done elegantly and efficiently using Differential Algebra (DA or TPSA) [5]
- This opens the possibility to fully exploit TPSA and output maps that actually depends on any arbitrary element (e.g. sextupoles, quads...)  $\Rightarrow$  **PTC knob**



- Maps at any given order are obtained with PTC (used drift-kick-drift model and 2<sup>nd</sup> order integrator with 10 integration steps)
- Developed a simple **cython** routine to read and used these maps for tracking (for doing many loops python is definitely not a good idea!)
- For the model validation used always 150 particles with different initial conditions (in  $x$  and  $\delta_p$ )  $\Rightarrow$  compared with PTC internal tracking (element-by-element) using the maximum relative error of the horizontal action:

$$\frac{\Delta J_x}{J_x} \equiv \frac{J_{x,maps} - J_{x,exact}}{J_{x,exact}} \quad (3)$$

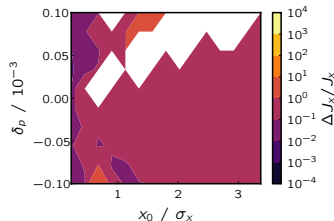
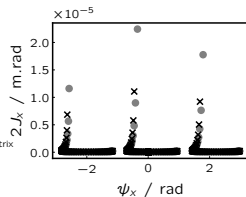
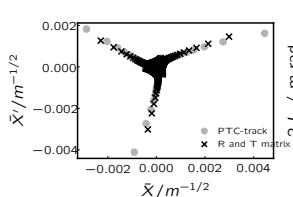
- In the following part, we will be focusing on comparing 2 scenarios:
  - Left: standard slow extraction (only sextupoles)
  - Right: "separatrix folding" slow extraction (more harmonic sextupoles and many octupoles)





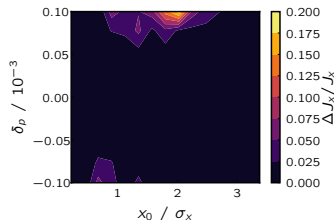
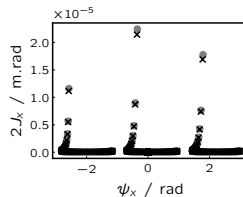
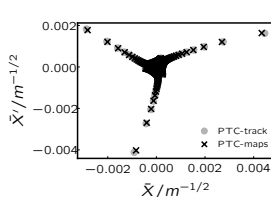
- The first scenario is the classic SPS slow extraction
- Relatively strong harmonic sextupoles, chromaticity close to natural ( $\xi_x = -1$ )
- As expected, using 2<sup>nd</sup> order maps does a very poor job at reproducing element-by-element tracking results...unusable!

2<sup>nd</sup> order



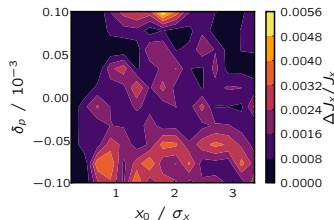
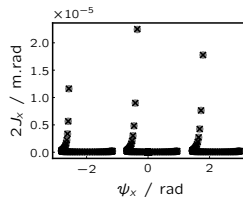
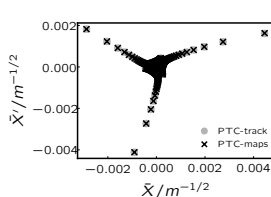
- The first scenario is the classic SPS slow extraction
- Relatively strong harmonic sextupoles, chromaticity close to natural ( $\xi_x = -1$ )
- As expected, using 2<sup>nd</sup> order maps does a very poor job at reproducing element-by-element tracking results...unusable!
- Going to 3<sup>rd</sup> order already improves drastically the situation

## 3<sup>rd</sup> order



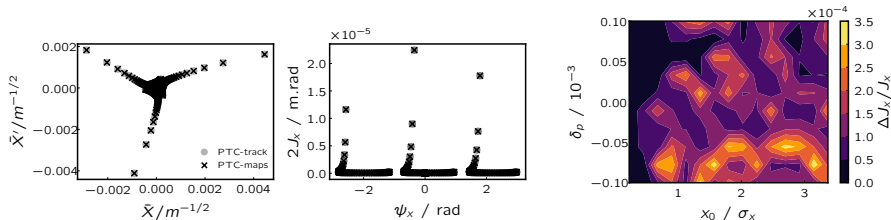
- The first scenario is the classic SPS slow extraction
- Relatively strong harmonic sextupoles, chromaticity close to natural ( $\xi_x = -1$ )
- As expected, using 2<sup>nd</sup> order maps does a very poor job at reproducing element-by-element tracking results...unusable!
- Going to 3<sup>rd</sup> order already improves drastically the situation
- From the 4<sup>th</sup> order there is almost no change anymore...converged!

4<sup>th</sup> order

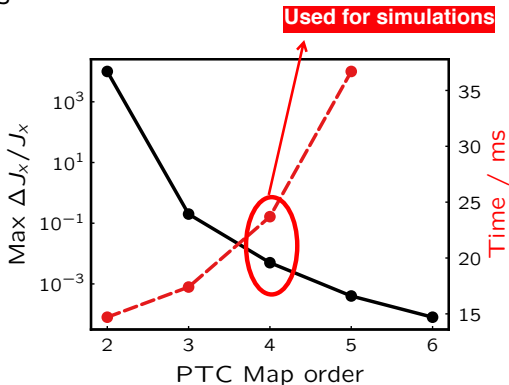


- The first scenario is the classic SPS slow extraction
- Relatively strong harmonic sextupoles, chromaticity close to natural ( $\xi_x = -1$ )
- As expected, using 2<sup>nd</sup> order maps does a very poor job at reproducing element-by-element tracking results...unusable!
- Going to 3<sup>rd</sup> order already improves drastically the situation
- From the 4<sup>th</sup> order there is almost no change anymore...converged!

5<sup>th</sup> order

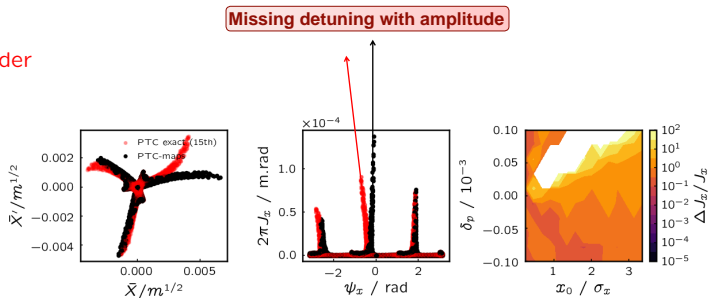


- Of course, higher the order, larger the time needed for tracking
- Time in the plot represents tracking of 1 particles for 300 turns



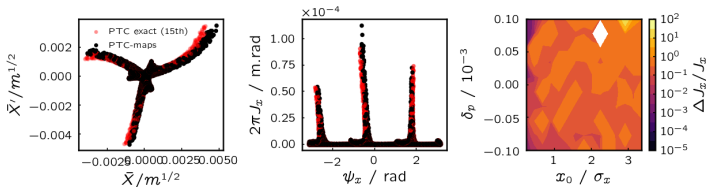
- For the separatrix folding case, the convergence is more complicated
- Splitting the SPS in only 4 parts (also not evenly...) really does not work for reasonable orders
- As expected, at the second order is very instructive to notice how **detuning with amplitude** is not taken into account

2<sup>nd</sup> order



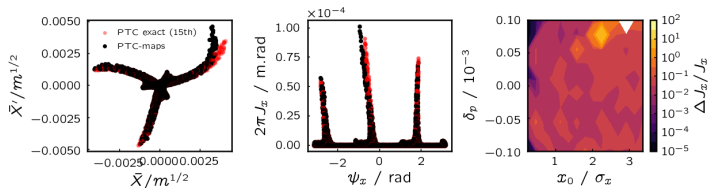
- For the separatrix folding case, the convergence is more complicated
- Splitting the SPS in only 4 parts (also not evenly...) really does not work for reasonable orders
- As expected, at the second order is very instructive to notice how detuning with amplitude is not taken into account

3<sup>rd</sup> order



- For the separatrix folding case, the convergence is more complicated
- Splitting the SPS in only 4 parts (also not evenly...) really does not work for reasonable orders
- As expected, at the second order is very instructive to notice how detuning with amplitude is not taken into account

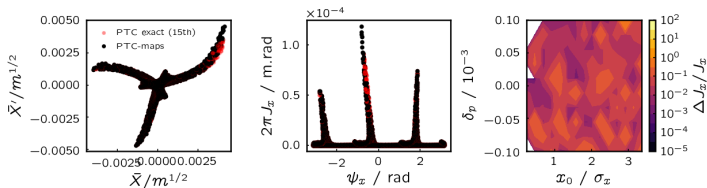
4<sup>th</sup> order





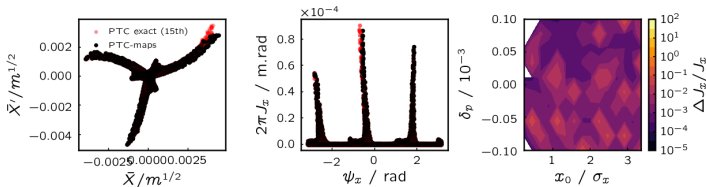
- For the separatrix folding case, the convergence is more complicated
- Splitting the SPS in only 4 parts (also not evenly...) really does not work for reasonable orders
- As expected, at the second order is very instructive to notice how detuning with amplitude is not taken into account

5<sup>th</sup> order



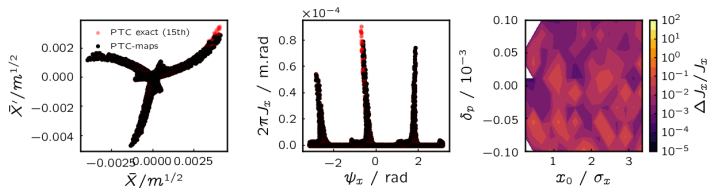
- For the separatrix folding case, the convergence is more complicated
- Splitting the SPS in only 4 parts (also not evenly...) really does not work for reasonable orders
- As expected, at the second order is very instructive to notice how detuning with amplitude is not taken into account

6<sup>th</sup> order



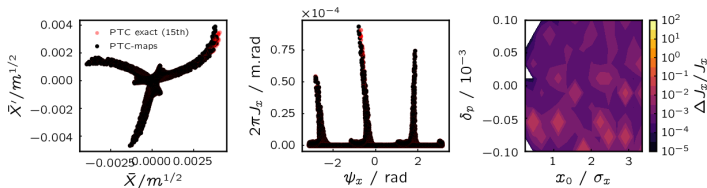
- For the separatrix folding case, the convergence is more complicated
- Splitting the SPS in only 4 parts (also not evenly...) really does not work for reasonable orders
- As expected, at the second order is very instructive to notice how detuning with amplitude is not taken into account

7<sup>th</sup> order



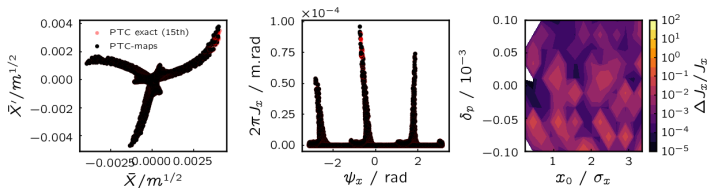
- For the separatrix folding case, the convergence is more complicated
- Splitting the SPS in only 4 parts (also not evenly...) really does not work for reasonable orders
- As expected, at the second order is very instructive to notice how detuning with amplitude is not taken into account
- Only at the 8<sup>th</sup> order we start to see some convergence...not very practical! Each single map contains 671 coefficients...
- Tracking for 300 turns 150 particles takes about 15 s - way too long!

8<sup>th</sup> order

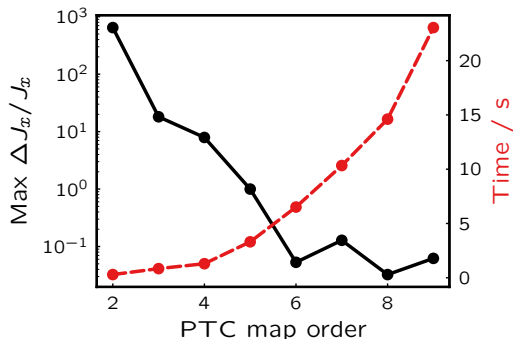


- For the separatrix folding case, the convergence is more complicated
- Splitting the SPS in only 4 parts (also not evenly...) really does not work for reasonable orders
- As expected, at the second order is very instructive to notice how detuning with amplitude is not taken into account
- Only at the 8<sup>th</sup> order we start to see some convergence...not very practical! Each single map contains 671 coefficients...
- Tracking for 300 turns 150 particles takes about 15 s - way too long!

9<sup>th</sup> order

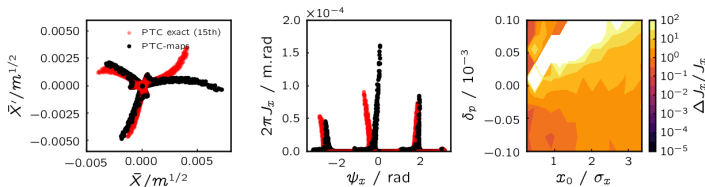


- Eventually it converged to maximum error of a few percent, but this is still poorer than what shown for the normal slow extraction case
- Different integration methods have been tried, as well as different integrator orders - no significant change
- It was then clear that the problem is in the number of maps used (mainly in the number of non-linear elements included in one map)



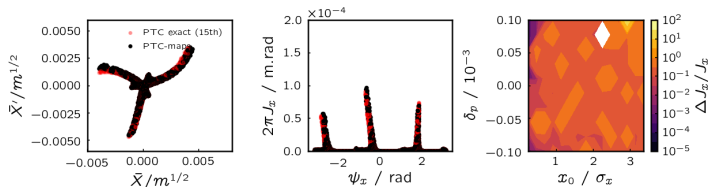
- As said, to improve the simulation accuracy, we need to slice the lattice in more sub-sectors
- Increasing the number of maps, keeping the order constant, will make the simulation time increase linearly

2<sup>nd</sup> order



- As said, to improve the simulation accuracy, we need to slice the lattice in more sub-sectors
- Increasing the number of maps, keeping the order constant, will make the simulation time increase linearly
- For example, we can see that with 8 maps the situation is drastically improved

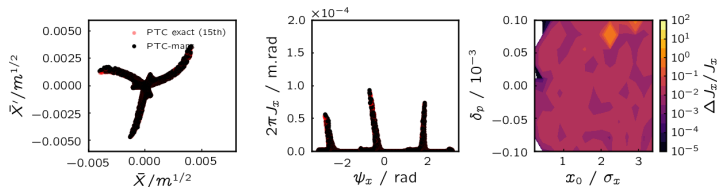
3<sup>rd</sup> order





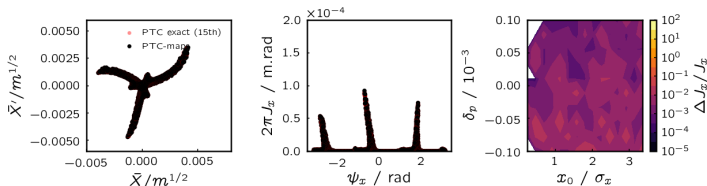
- As said, to improve the simulation accuracy, we need to slice the lattice in more sub-sectors
- Increasing the number of maps, keeping the order constant, will make the simulation time increase linearly
- For example, we can see that with 8 maps the situation is drastically improved
- Already at the 4<sup>th</sup> order, although still some high errors in some isolated cases, the accuracy is now comparable with the standard slow extraction case

4<sup>th</sup> order



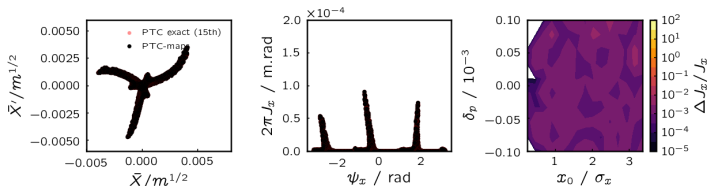
- As said, to improve the simulation accuracy, we need to slice the lattice in more sub-sectors
- Increasing the number of maps, keeping the order constant, will make the simulation time increase linearly
- For example, we can see that with 8 maps the situation is drastically improved
- Already at the 4<sup>th</sup> order, although still some high errors in some isolated cases, the accuracy is now comparable with the standard slow extraction case

5<sup>th</sup> order

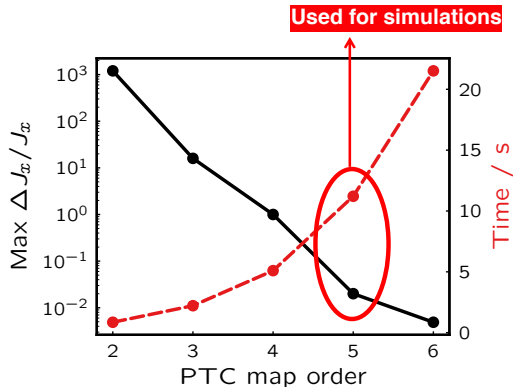


- As said, to improve the simulation accuracy, we need to slice the lattice in more sub-sectors
- Increasing the number of maps, keeping the order constant, will make the simulation time increase linearly
- For example, we can see that with 8 maps the situation is drastically improved
- Already at the 4<sup>th</sup> order, although still some high errors in some isolated cases, the accuracy is now comparable with the standard slow extraction case

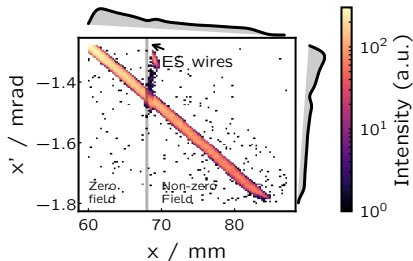
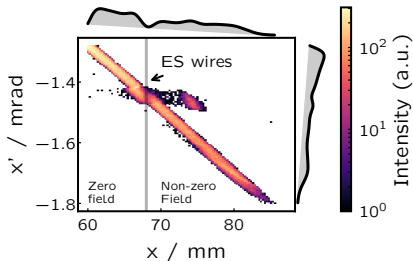
6<sup>th</sup> order



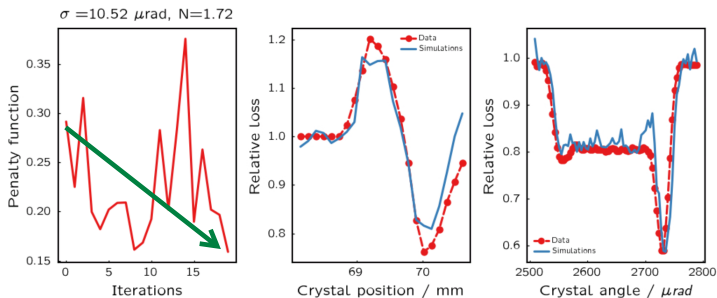
- Summarising, we manage to get accurate and fast simulations also for the case with octupoles
- Optimisation still possible - need to try to split better long sectors (max length now is 1200 m). It can be reduced



- A first example is the simulation of crystal shadowing
- In this case, 4 maps truncated at the 4<sup>th</sup> order have been used

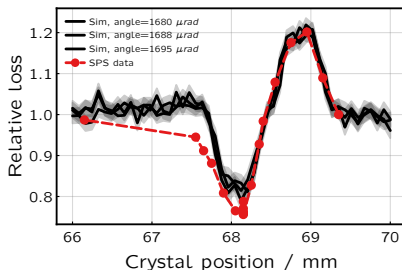
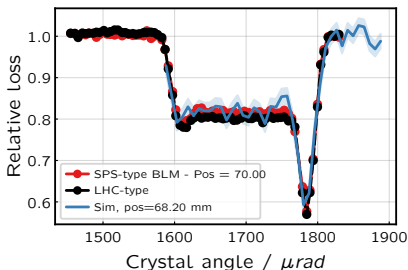


- A first example is the simulation of crystal shadowing
- In this case, 4 maps truncated at the 4<sup>th</sup> order have been used
- Also, these fast simulations are usable for automatic optimisation as done for the crystal shadowing case

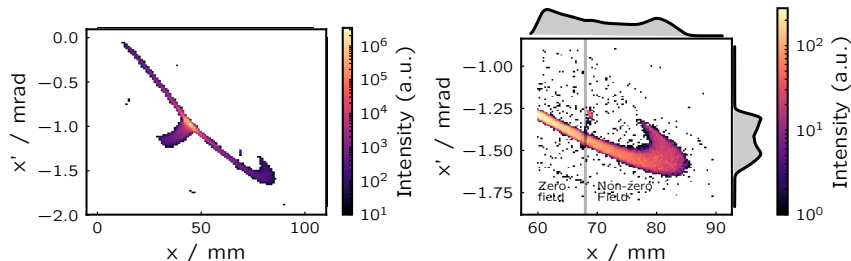


- A first example is the simulation of crystal shadowing
- In this case, 4 maps truncated at the 4<sup>th</sup> order have been used
- Also, these fast simulations are usable for automatic optimisation as done for the crystal shadowing case

Pretty good results comparing with data

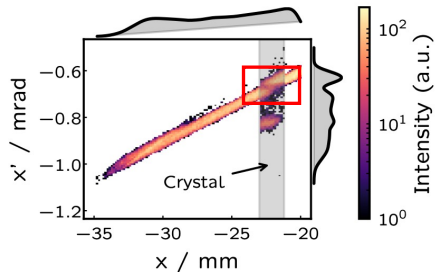
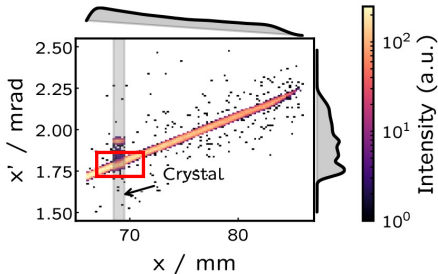


- A first example is the simulation of crystal shadowing
- In this case, 4 maps truncated at the 4<sup>th</sup> order have been used
- Also, these fast simulations are usable for automatic optimisation as done for the crystal shadowing case
- Then, the natural extension of this is the simulation of crystal shadowing together with the separatrix folding via octupoles
- Here 8 maps truncated at the 5<sup>th</sup> order have been used

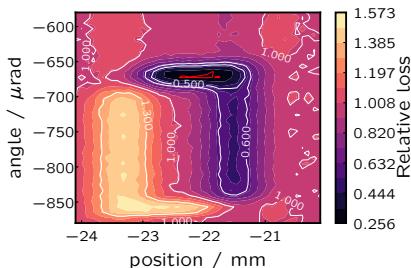
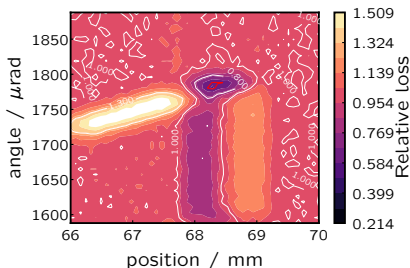




- A further speed up can be obtained reducing the number of turns and properly selecting the particles to track
- No need to track particle around the core!
- No need to track particle from their stable position...one can “condition” simulations first and then only select the phase-space area of interest and ad-hoc populate it
- As imaginable, now the simulations of the full dead-time is not needed anymore, hence 3-6 turns are necessary to make all circulating particles being extracted
- In these cases, also scattering inside material was considered (pycollimate)



- A further speed up can be obtained reducing the number of turns and properly selecting the particles to track
- No need to track particle around the core!
- No need to track particle from their stable position...one can “condition” simulations first and then only select the phase-space area of interest and ad-hoc populate it
- As imaginable, now the simulations of the full dead-time is not needed anymore, hence 3-6 turns are necessary to make all circulating particles being extracted
- In these cases, also scattering inside material was considered (pycollimate)



- Slow extraction simulations, in most cases, require tracking to evaluate performance
- For the SPS slow extraction, we developed tracking simulations based on a few high order maps that describe the whole lattice  $\Rightarrow$  maps obtained from PTC
- This gave access to to very accurate (compared with pure element-by-element tracking) and very fast tracking simulations
- For the different scenarios under analysis, accuracy of the simulations was tested, number and order of maps were chosen to maximise accuracy and minimise simulation time
- Already well established procedure for crystal shadowing simulations (also benchmarked with measurements) and firsts tests done for crystal shadowing with octupoles
- We will look next at full characterisation of parameter space for separatrix folding technique and then combine it with crystals
- Exploit automatic optimisers to evaluate best parameters combination
- Use simulations to test different numerical optimisers and then deploy it in operation

- [1] F. Forest, et al., "Introduction to the Polymorphic Tracking Code", CERNâSLâ2002â044 (AP)
- [2] E. Forest, "From Tracking Code to Analysis", Springer, ISBN 978-4-431-55802-6, 2016
- [3] H. Yoshida, Phys. Lett A150 (1990) 262
- [4] W. Herr, E. Forest, "Non-linear Dynamics in Accelerators", CERN Accelerator School, Royal Holloway University of London, 2017
- [5] M. Berz, Particle Accelerators 24, (1989) 109